

Task Brief: BDD Test Case Support

[Customer Asks](#)

[Executive Brief:](#)

[What is the problem?](#)

[Pre-read on BDD](#) IMPORTANT

[What is BDD \(Behavior-Driven Development\)?](#)

[Glossary](#)

[Key Workflows](#)

[Example BDD Test Case](#)

[Who is facing this problem?](#)

[Why do we want to solve this problem?](#)

[What is the current behavior?](#)

[How do our competitors solve this?](#)

[Solution](#)

[BDD-Steps mapping while managing test runs](#)

[Single Test Case Export](#)

[2. Multiple Test Cases Export](#)

[3. Export Workflow Behavior](#)

[4. Feature File Structure](#)

[1. Import Option in Test Case Listing View](#)

[2. File Selection and Upload Restrictions](#)

[3. Test Case Creation from Feature File](#)

[4. Folder Selection and Organization](#)

[7. Import Processing and Validation](#)

[Exclusions in Phase 1](#) TO BE EXPLORED FURTHER LATER

[1. Field Mapping Logic](#)

[2. Conditional Field Availability](#)

[3. Template Naming Updates](#)

[RBAC](#)

[Scenario Handling](#)

[Design Wireframe](#)

[Phase 2](#)

[PRFAQ / Newsletter update](#)

[Non-functional Requirements](#)

[GTM Details](#) WIP

[Instrumentation](#)

[Success Metrics](#)

[References / Key MOMs](#)

[Separate Fields VS one Gherkin Field](#)

Task	Link	Status
Scoping		WIP
Dependency teams		NONE
Design	 BDD	WIP

Design Stack dependency		NONE
Walkthrough		TO BE UPDATED Sign in to access Google Drive Folder
Effort Estimate		TO DO
Instrumentation		TO DO
Documentation		REQUIRED

Customer Asks ↗

Feature request	Customer tickets
BDD support for Test Cases & Runs in Test Management	TM-8390: [Heady.io] BDD Support for Importing and Creating Test Cases CLOSED PRDFBK-3710: AI test authoring using Gherkin Feature files. NOT YET PLANNED CTO-2281: Detailed test case generation of Gherkin format NEW ITEM TM-9602: [Holidu] Writing test cases in Gherkin syntax and exporting them as BDD feature files CLOSED

Executive Brief: ↗

Executive Brief: BDD test case support for Test Management

Feature Parity Sheet Name:

Problem statement

Currently, Test Management allows users to create test cases in two formats - Text or Steps. As per user request, we need to support:

1. Phase 1: The user should be able to:

- a. create test cases in Gherkin format (BDD)
 - i. Functionalities like Feature, Scenario Outline, Examples etc. supported
 - ii. The tests should appear with proper colour formatting
- b. Execute BDD Test cases - Manually as well as via Automation (BDD Json/JUnit XML)
- c. export test cases in .csv or .feature file format
- d. Importing test cases using .feature files or .csv file
- e. import BDD test cases from Zephyr Quick Import, test rail quick import

2. Phase 2:

- a. BDD Test case generation using AI
- b. Users should be able to import test cases from Xray in BDD format
- c. Automatically generate BDD test cases from the tests run through Test Observability
- d. APIs to import & export BDD test cases

e. Integrate their CI/CD pipeline and directly import test cases from the same

- Customer asks documented
- Competitive Study completed
- Idea Validation (Squad 2)
- Task table updated (eng dependencies, doc,
- Phases / Squads identified
- Launch steps identified (does this impact other customers, will this need a GTM action, etc)
- Eng collaboration
- Dependencies with other teams identified
- T-shirt sizing and scoping discussion with engineering
- Design & QA heads up (post exec review)

Solution:

1. **Phase 1:** BDD template that allows users to:

- a. create test cases in Gherkin format
 - i. Integrate Gherkin authoring format in the text editor for TM
 - ii. Introduce a third option under the 'Type' dropdown
 - iii. Introduce a data table option (EXAMPLES) on the top right corner of the description box
- b. Execution - Both manual and automated execution via reports to be given
- c. import BDD test cases from Test Rail using the current 'Import from TestRail/Xray' feature
 - i. Identify template using the 'Template' field in TestRail/xray. If its BDD, then follow steps ii-iv.
 - ii. Each 'Scenario' in Gherkin format is treated as a separate test case within test management
 - iii. Map the following fields accordingly:
 - 1. Preconditions → Preconditions
 - 2. Status → Status
 - 3. Type → Type
 - 4. Priority → Priority
 - 5. Automation Type → ?
 - iv. Rest of the fields can be left as 'null' and the user shall be able to edit them basis their needs.
- d. export test cases in .csv or .feature file format
 - i. introduce export capabilities at both these areas - top of the panel and to the right side of each test case
- e. Importing test cases using .feature files or .csv file

GTM Actions

Following are the channels that the feature announcements would be conducted:

1. Email
2. Beamer
3. Newsletter
4. Landing Page Update
5. Slack Sales Channel Announcement
6. LinkedIn
7. Update Sales collaterals
 - o Battlecards
 - o Pitch Deck
 - o Sell Sheet

What is the problem?

Currently, Test Management allows users to create test cases in two formats - Text or Steps. (More details [here](#)) Some of our users have requested for support for BDD template while creating/importing/exporting test cases.

Pre-read on BDD IMPORTANT

- What is BDD - Please read this short summary to develop a basic understanding essential for the requirements

What is BDD (Behavior-Driven Development)?

Behavior-Driven Development (BDD) is a collaborative approach to software development where test cases are written in plain language to describe expected system behavior. BDD uses a structured format that can be executed by automation tools, making it accessible to both technical and non-technical stakeholders.

Glossary

1. Feature:

- A high-level description of a functionality or behavior of the system.
- Example: "*Booking a Flight*"

2. Scenario:

- A specific example or use case within a feature that describes a behavior.
- Example: "*Booking a one-way flight with a valid credit card*"

3. Scenario Outline:

- A template for a scenario that allows testing multiple examples with different input values.
- Example: Testing multiple payment methods for a flight booking.

4. Examples:

- Tabular data used in Scenario Outlines to define various input/output combinations.
- Example Table:

1	Payment Method Status
2	Credit Card Approved
3	Debit Card Declined

5. Given-When-Then:

- The syntax used to define the steps of a scenario:
 - Given:** Defines the initial context (preconditions).
 - When:** Describes the action taken.
 - Then:** Specifies the expected outcome.
- Example:

1	Given the user is logged in
2	When the user books a flight
3	Then the booking confirmation is displayed

6. Feature File:

- A file format (e.g., `.feature`) where BDD test cases are written. It contains all features, scenarios, and steps in plain text.

7. BDD Automation Tools:

- Tools like **Cucumber**, **SpecFlow**, and **Behave** execute `.feature` files by mapping steps to code.

8. BDD JSON Report:

- A report format generated by automation tools that captures the results of running `.feature` files, including pass/fail status for each scenario and example.

9. JUnit XML Report:

- A widely used report format for automation results, which can also be used to map BDD results back into a test management tool.

Key Workflows

1. Authoring BDD Test Cases:

- Use a structured editor to write Features, Scenarios, and steps in the tool.
- Add Scenario Outlines with Examples for parameterized testing.

2. Exporting BDD Test Cases:

- Generate `.feature` files for execution in automation tools like Cucumber.

3. Importing BDD Test Cases:

- Upload `.feature` files to the tool, which parses and structures them into Features, Scenarios, and Examples.

4. Manual Execution:

- Execute BDD scenarios or examples manually in Test Runs.
- Update results (Pass/Fail/Blocked) for each scenario or example.

5. Automated Execution:

- Run `.feature` files in automation tools.
- Import BDD JSON or JUnit XML reports to map scenario results back to the tool.

Example BDD Test Case

Feature: Booking a Flight

Scenario Outline: Booking with different payment methods

- **Given** the user is on the payment page
- **When** they pay using `<Payment Method>`
- **Then** the payment is `<Status>`

Examples:

1	Payment Method Status
2	Credit Card Approved
3	Debit Card Declined

Who is facing this problem?

All the users who want to migrate to TM from our competition (Test Rail). Few prospects also had requested for this feature. It was initially a deal blocker for an account [# TM-8390: \[Heady.io\] BDD Support for Importing and Creating Test Cases !\[\]\(e119fc79c8f448683d20ba4c873025a2_img.jpg\)](#).

Why do we want to solve this problem?

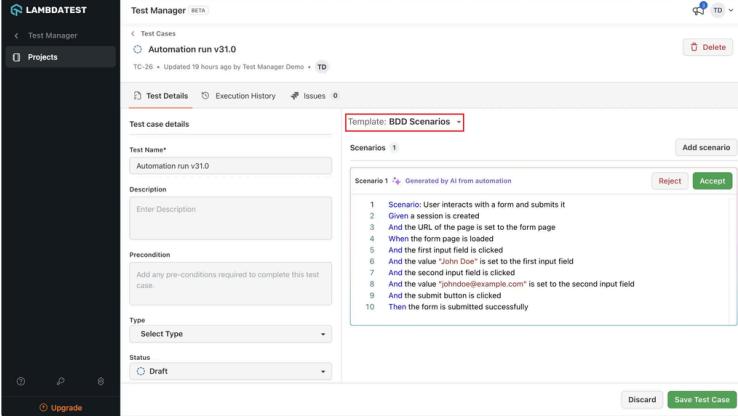
- This is a feature request from an existing customer.
- This brings us at par in terms of feature parity across our competitors.

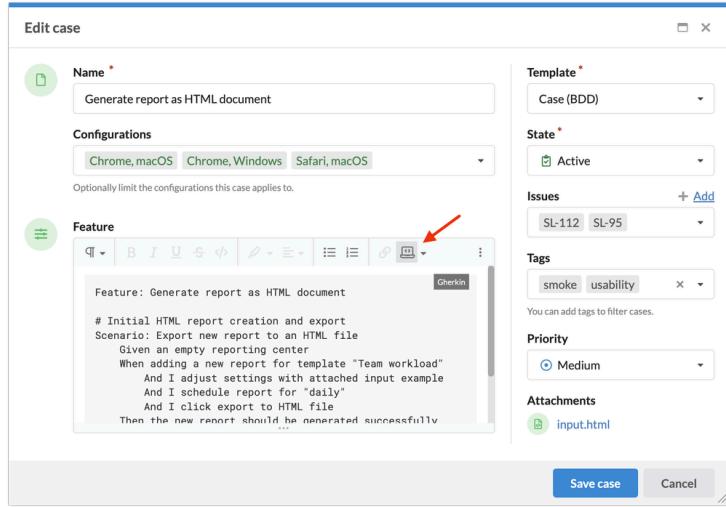
What is the current behavior?

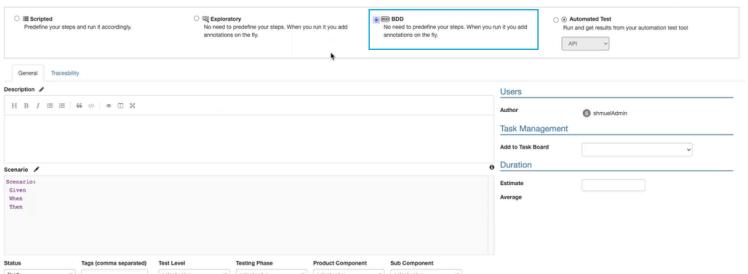
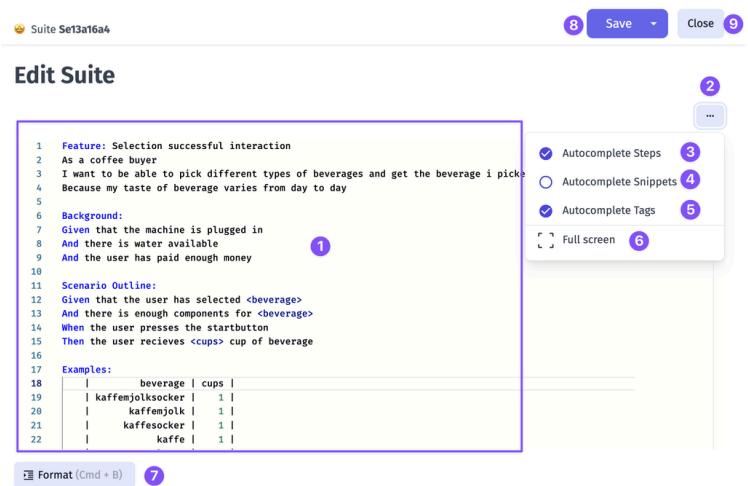
- Users can run test cases in two formats - Text or Steps. (More details [here](#))
- We currently support uploading BDD-JSON based cucumber reports to Test Management. The tests imported through this route are imported in 'Text' format.

- Test Management also integrates tests directly from Test Observability. The tests imported through this route are imported in 'Text' format.

How do our competitors solve this? ↗

Competitor	Description
Smartbear	<p>PRESENT</p> <ol style="list-style-type: none"> 1. Provides a very intuitive way of writing test cases in BDD. 2. Has color formatting for keywords & parameters 3. has a separate data table for EXAMPLES 4. considers each line under a scenario as test step <p>Writing scenarios with Gherkin syntax CucumberStudio Documentation</p>
TestRail	<p>PRESENT</p> <ul style="list-style-type: none"> • If you select a test case that has the BDD template format and scenarios in it, you can run the BDD scenarios manually the same way you would run a test case that uses steps in it. (More details here). Also supports “Examples” table in the scenario • Export BDD test case to .feature file: (More details here) This is supported by API endpoints as well. • Import .feature file as test cases: (More details here) This is supported by API endpoints as well.
LambdaTest (Test Manager)	<p>PRESENT</p> <p>BDD scenarios are automatically generated (using AI) from the command logs for the automated tests executed.</p> 
Qase	<p>PRESENT</p> <p>Two types of Test Case steps supported: Classic & Gherkin</p>

	<p>Test Case Steps Gherkin ▾</p> <pre>Given the website is fully loaded. When you are prompted to enter login credentials. Then click on signup and enter your name, username and password. And tick the 'I agree to terms and conditions' box. But do not tick the 'remember my password' box.</pre> <p>Raw <input checked="" type="checkbox"/> Steps</p>
TestMo	<p>PRESENT (Shows a separate bdd template in docs, not there in free trial. The BDD format syntax is here)</p> <p>Testmo supports behavior-driven development (BDD) and Gherkin syntax to manage your software tests.</p> <ul style="list-style-type: none"> • Use popular BDD tools such as Cucumber, Robot Framework, Jasmine, RSpec and many more • Support for Gherkin syntax highlighting for your manual test cases and test runs • Import & Export .feature files  <p>More details here</p>

Qmetry	PRESENT  <p>Efficiency and Improved Performance</p> <p>AI-powered defect detection and enhanced test management systems swiftly address bottlenecks to ensure product quality and efficiency, integrating seamlessly with Jira for streamlined test authoring and user-friendly experience with native BDD support.</p>
PractiTest	PRESENT  <p>More details here</p>
Testomat	PRESENT  <p>More details here</p>

Solution

BDD-Steps mapping while managing test runs

The Gherkin template to Steps Template map should look like the following:

Each Scenario would be mapped to a test case. A test case will only have one scenario field with the Gherkin editor. The User would be able to add a result to it.

1. In the **backend**,
 - a. each line item / Scenario will be considered a single step
 - b. The EXAMPLES (data tables) will be referred for each line item to create a test & populate them simultaneously as part of the folder (Similar way what we have for configuration) - **PHASE 2**
 - i. For ex: a scenario outline that refers to a data table with 3 rows can create 3 individual test cases in the test run view
2. The same UI for adding a result for Text template to be followed.

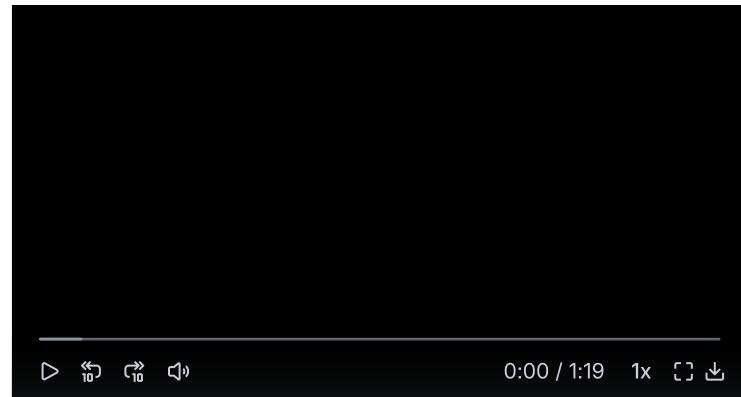
	Phase	Jobs to be done	Implementation
1	1	I should be able to create test cases using the 'Create Test Cases' in the Gherkin format (Authoring experience)	<p>Add third option for users to choose BDD template while Creating Test Cases</p> <p>1. Image Ref Shown, BDD should be added in the list</p>  <p>2. The UI will have the following fields for the user:</p> <ol style="list-style-type: none"> a. Title: To be reused (Mandatory) b. Feature field: Here the user outlines the specific feature under test - This should be gherkin editor, with the default value of the folder name, User can change this later . A tooltip near this to inform that the folder name is taken as default feature name and can be changed (Mandatory) c. Description: Same field to be reused d. Background: similar to Precondition field - Gherkin Editor e. Scenario: The following field should let the user add scenarios / scenario outlines. The user can also add EXAMPLES in the same editor. (Mandatory) <p>BDD Editor field support is live from DS end</p>

		<p>Code Editor</p> <pre> 18 When I proceed to checkout 19 And I fill in my shipping information 20 And I select "Express Shipping" 21 And I enter valid payment details 22 Then my order should be confirmed 23 And I should receive an email confirmation 24 And my total should be \$119.97 25 26 Scenario Outline: Apply different discount codes 27 Given I have a "<product>" in my cart 28 When I apply the discount code "<code>" 29 Then the total price should be reduced by <percentage>% 30 31 Examples: 32 product code percentage 33 T-shirt SUMMER10 10 34 Jeans SALE20 20 35 Sneakers SPECIAL30 30 </pre>
		<ol style="list-style-type: none"> 4. This text editor should support Gherkin syntax highlighting shown above 5. The fields on the right side (Status, priority, tags Req etc) would remain the same. 6. The user will still be able to add custom fields. 7. Impact on custom templates <ul style="list-style-type: none"> a. Scenario, Feature, background fields should be available in custom templates also. b. BDD template should also be a system template c. Changes made via template customization should reflect here with (Feature, Scenario, Title, Background being the fields that cannot be changed on the left side) d. While creating a custom template, a user selects scenario and not steps - user should be allowed to do so
2	1	<p>I should be able to Import test cases from TestRail,Xray & Zephyr into Test Management (Zephyr to be the priority due to DB)</p> <p>Currently TestRail & zephyr supports BDD format with the following basic fields:</p> <p>For TestRail, which supports multiple test scenarios in one case, we will follow the same behavior which is being done currently in quick import, but if the template is BDD, by default we should change the code snippet to gherkin in the steps template</p> <p>Zephyr</p> <ol style="list-style-type: none"> 1. Title - to be mapped to Title in TM, Feature in TM 2. objective - to be mapped to Description in TM 3. Preconditions - To be mapped to Preconditions (For all such cases - we will create the preconditions field) - we will not be showing this field in the create view though. 4. Scenario <p>The new import methodology should work exactly in the same manner with a slight difference:</p> <ol style="list-style-type: none"> 1. Identify template using the 'Template' field in TestRail/Xray. if its BDD, then follow: steps 2 - 7. 2. Each 'Scenario' in Gherkin format is treated as a separate test case within test management For Zephyr, 3. Preconditions → Background 4. The TM field of feature should have the folder name as its default value

5. Status → Status
6. Type → Type
7. Priority → Priority
8. Automation Type → ?

Rest of the fields similar mapping to be followed as we have in quick import as of now

✓ Current Workflow



0:00 / 1:19 1x []

3	1	<p>I should be able to export BDD test cases as .feature files</p>	<p>Once the test case is created the user should be able to export test cases as .feature files</p> <p>The workflow remains the same. User selects the test case and clicks 'Export'. The export button should have 2 options - 'Export as CSV' & 'Export as .feature file'.</p> <p>Validations</p> <p>No validation while writing: Users can enter any text in Feature, Background, and Scenario fields. Skipping validation reduces engineering effort since fields are already separate.</p> <ol style="list-style-type: none"> 1. Trim leading spaces: Remove any leading white spaces before processing. 2. Ensure correct Gherkin keyword per field: <ul style="list-style-type: none"> ◦ Scenario field → Must start with <code>Scenario:</code> (detect and use the first occurrence). ◦ Feature field → Must start with <code>Feature:</code> (detect and use the first occurrence). ◦ Background field → Must start with <code>Background:</code> (detect and use the first occurrence). ◦ If missing, prepend the correct keyword before exporting. <p>Single Test Case Export</p> <ul style="list-style-type: none"> • When a user selects one BDD test case and chooses "Export as .feature file", the generated <code>.feature</code> file will include: <ul style="list-style-type: none"> ◦ Feature → Taken directly from the test case's Feature field. ◦ Background (Optional) → Included only if present in the test case. ◦ Scenario → The test case's Scenario will be exported as the only Scenario in the <code>.feature</code> file.
---	---	--	--

2. Multiple Test Cases Export

- If **multiple BDD test cases** are selected for export, the system will check if the **Feature or Background fields differ** among them.
- A checkbox is given to the user which asks them if they want to combine all these test cases under a single feature. (Default value -not Selected). If this is not selected, export all feature files in a zip file.
- If the above is selected
 - If **all test cases have the same Feature & Background**, the `.feature` file is generated using those values.
 - If **Feature or Background fields differ**, a modal appears requiring the user to:
 - Telling the user that multiple feature/background names are found, please enter the fields values for the final export
 - Enter a **Feature** name.
 - Enter a **Background** (optional).
 - Once entered, all selected test cases will be exported under the provided Feature & Background.

3. Export Workflow Behavior

- The "Export as `.feature` file" option is available **only when BDD test cases are selected**.
- If a non-BDD test case is selected, the option is **hidden**.
- If both BDD and non-BDD test cases are selected, the non-BDD test cases are **ignored**, and only BDD test cases are exported.

4. Feature File Structure

- **File name:**
 - a. For Single TC export: `<FeatureName>_<TCID>.feature` Eg - Login_TC-123.feature
 - b. For Combined TCs in a single file: `<FeatureName>.feature`
 - c. For Zip File - `BDD_Feature_files_<Date (dd-mm-yy)>` Eg- `BDD_Feature_Files_18-02-25.zip`
- **Feature:** User-defined or taken from the test case(s).
- **Background:** User-defined (if provided) or taken from test case(s) (if consistent).
- **Scenarios:** Tc's Scenario within the `.feature` file.

4	1	<p>I should be able to import test cases in BDD format using <code>.feature</code> files</p>  <p>1. Import Option in Test Case Listing View </p> <ul style="list-style-type: none"> • The existing Import button for CSV to continue • Selecting Import will open the Import Modal. • Users would now be able to add feature files also <p>2. File Selection and Upload Restrictions </p> <ul style="list-style-type: none"> • Users can upload one/multiple .feature files at a time. (Same UI for CSV to be followed) • Maximum file size: 10MB. • Users must select a Root Folder before proceeding (Same UI for to be followed)
---	---	---

- If the file exceeds the limit or is in an invalid format, an error message will be displayed.

3. Test Case Creation from Feature File

- The system will parse the uploaded `.feature` file and create **individual test cases** for each Scenario/Scenario Outline inside it.
- Field Mapping: BDD template to be used
 - Test Case Title** → Scenario/Scenario Outline name from the `.feature` file.
 - Feature** → Feature field from the `.feature` file.
 - Background** (if present) → Applied to all test cases from the same `.feature` file.
 - Steps (Given-When-Then)** → Mapped inside the test case's Scenario field.
 - For rest of the fields in the template give the default values
- If a **Scenario has no name**, the **Feature name will be used as the Title**.

4. Folder Selection and Organization

- Users must select a **Root Folder** for the imported test cases.
- All test cases from the Feature file will be placed **directly inside the selected Root Folder**.
- No automatic folder creation based on Feature name.

5. Multiple File Upload Support

- Users should be able to upload multiple files at once instead of being restricted to a single file.
- The UI should support drag-and-drop and multi-select file upload.

6. Automatic Folder Creation Based on Feature Name

- A checkbox should be provided to enable/disable automatic folder creation.
 - If selected, a new folder should be created inside the root folder based on the feature name extracted from the uploaded file(s).
 - If not selected, files should be uploaded without automatic folder structuring.

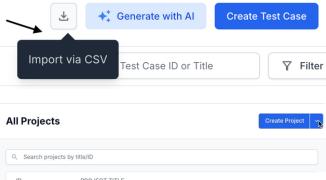
7. Import Processing and Validation

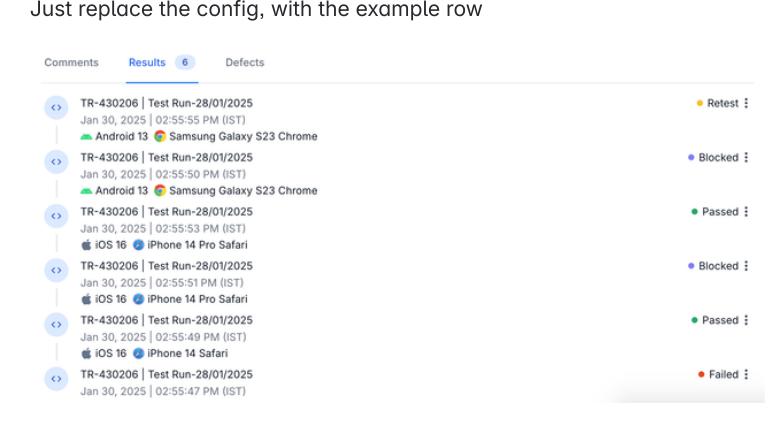
- If no Root Folder is selected, the user will be prompted to select one.
- If the `.feature` file format is invalid, an error message will be displayed.
- If a Background is present, it will be applied to all test cases from the same Feature.
- No updates will be made to existing test cases in this phase.
- If user clicks on cancel, then the entire import stops with a rollback.

Validations & Error Messages

Validation Rule	Description	Error Message
File Type Check	Only <code>.feature</code> files allowed	Invalid file type. Only <code>.feature</code> files are supported.
File Size Limit	Each file must be \leq 10 MB	File size exceeds 10 MB limit.

		Missing Feature	No Feature: keyword found	Missing 'Feature:' keyword. Please include a valid feature definition.
		No Scenario Found	File doesn't have any Scenario: or Scenario Outline:	At least one 'Scenario:' or 'Scenario Outline:' is required.
		Max File Count Exceeded	More than 20 files selected	You can upload up to 20 feature files at once.
Exclusions in Phase 1 <small>TO BE EXPLORED FURTHER LATER</small> 				
<ul style="list-style-type: none"> • No test case updates; 				
5	1	I should be able to export BDD test cases in CSV format	<ul style="list-style-type: none"> • The users should be able to import or export BDD test cases in CSV format just like they do it for other step-based templates. • Refer to sample_bdd.csv for field mapping, new fields "Scenario", background And Feature has been added for BDD based test cases. 	

6	1	Automated Runs - BDD Json and Junit XML automation	<p>The current behaviour works for BDD Json and Junit XML already - needs to be verified and TC id tagging support to be added , In BDD Json,at a scenario level there would be a field inside tags for TC id. Eg - "tags": [{ "name": "@TCID_TC-102" },]</p> <p>Customer only has to add the tag in scenario level in the feature file and gherkin will include that tag in the json</p> <div style="border: 1px solid #ccc; padding: 10px;"> <p>▼ EG</p> <pre> 1 Feature: My Orders Page 2 3 As a Kering employee 4 I want to view my orders 5 So that I can manage them easily 6 7 Background: 8 Given I am a Kering Employee 9 And I am logged in 10 11 @TCID_TC-101 @Smoke 12 Scenario: Empty Orders 13 And I have not closed any order 14 When I reach My Orders Page 15 Then the page should show an empty state 16 17 @TCID_TC-102 @Regression 18 Scenario: With Closed Orders 19 And I have closed one or more orders 20 When I reach My Orders Page 21 Then the page should list those orders </pre> </div>
7	1	I should be able to import BDD test cases in CSV format (This should be supported by default BE to confirm)	 <ul style="list-style-type: none"> The same csv logic to be followed, User should be able to map to the BDD fields - Feature,Scenario,Background. User would be able to select the template as BDD - (Solved in custom templates) <p>1. Field Mapping Logic</p> <ul style="list-style-type: none"> When the template is set as BDD (either automapped or the user manually maps during field mapping): <ul style="list-style-type: none"> Fields Feature, Scenario, and Background will be available for mapping. These fields will be auto-mapped if the column names match. Manual mapping is supported if column names differ. <p>2. Conditional Field Availability</p> <ul style="list-style-type: none"> If at least one test case in the CSV has the template set to BDD, the fields Feature, Scenario, and Background will be shown in the mapping screen. However, mapping of these fields will only be applied to test cases where the template is BDD.

		<ul style="list-style-type: none"> For test cases with other templates: <ul style="list-style-type: none"> These three fields will be ignored during import. All other fields will follow the standard import logic. <p>3. Template Naming Updates </p> <ul style="list-style-type: none"> Rename "Gherkin BDD" to "BDD" in CSV exports. Rename "Test Case - BDD" to "BDD" in the template dropdown. Reason: To maintain consistency across the UI and CSV, and to support reliable auto-mapping. 	
8	2	<p>Example Support during execution</p> <p>1. Test Case Instantiation</p> <ul style="list-style-type: none"> For each row in the Examples Table of the Scenario field, a new test case instance should be created during execution, with the values from that row automatically replacing placeholders in the Scenario Outline. <p>2. Execution & Results Tracking</p> <ul style="list-style-type: none"> Each test case instance should be executed independently, and the results (Pass/Fail) for each example row should be recorded separately in the Test Runs view. <p>3. Parallels with Configurations</p> <ul style="list-style-type: none"> This Example-based execution will function similarly to how Configurations work in Test Management, where multiple test runs are created for different configurations. In Test case view, for a TC having multiple examples, the results section should show the results for each example in a similar way for configuration <p>✓ Image ref for the above</p> <p>Just replace the config, with the example row</p> 	
9	3	I should be able to generate Test Cases using AI	Users should be able to generate test cases in a specific format. Currently by default the test cases are generated in 'Text' template. TO BE CHECKED
10	3	I should be able to automatically generate test cases from the tests run through Test Observability	<p>1. Automatic Generation of Test Cases</p> <ol style="list-style-type: none"> Automated tests are automatically imported from Test Observability Users should be automatically be able to see the BDD formatted test cases once their test runs are reflected Test Steps are listed in Test Observability dashboard under the 'steps' tab. leverage this to import the same in the specified BDD format in Test Case Management.

11	3	I should be able to run API commands to import or export test cases	1. Support all the existing APIs to create, retrieve, update and delete a test cases 2. Bulk operations are supported for test cases
12	3	I should be able to import or export test reports	1. Uploading BDD based test report using curl commands and then generate test cases with them(BDD-JSON based report upload using Cucumber Browser Stack Docs) TO BE CHECKED
13	3	I should be able to integrate their CI/CD pipeline and directly import test cases from the same	This feature is already supported for existing test cases with the test cases being imported in text template. TO BE CHECKED
14	3	I should be able to import test cases from Zephyr in BDD format	
15	3	I should be able to import test cases from Xray in BDD format	
16	3	I should be able to import test cases from qtest in BDD format	

RBAC

All the users of TM should be able to use this feature unless this is feature fenced going forward. All the access permissions should follow suite as that of other test type in TM.

Scenario Handling

Scenario	Requirement

Design Wireframe

Phase 2

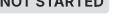
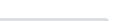
- ✓ Out of Scope Scenarios

PRFAQ / Newsletter update

Non-functional Requirements

GTM Details

CTA for all the following channel announcements would be 'View Documentation'

Channel	Requirements	CTA	Date	Status
Email	<ul style="list-style-type: none"> • Recipient List • Email Copy • Documentation Link 	<ul style="list-style-type: none"> • Primary: Get Started Free • Secondary: View Documentation 	TBD	
Beamer	<ul style="list-style-type: none"> • Design • Beamer Copy • Documentation Link 	<ul style="list-style-type: none"> • Primary: View Documentation 	TBD	
Newsletter	<ul style="list-style-type: none"> • Copy • Documentation Link 	<ul style="list-style-type: none"> • View Documentation 	TBD	
Landing Page Update	<ul style="list-style-type: none"> • Design • Copy • Documentation Link 	<ul style="list-style-type: none"> • Primary: Get Started Free • Secondary: View Documentation 	TBD	
Slack Sales Channel Announcement	<ul style="list-style-type: none"> • Copy • Documentation Link 	<ul style="list-style-type: none"> • View Documentation 	TBD	
Linkedin	<ul style="list-style-type: none"> • Social copy • Documentation Link 	<ul style="list-style-type: none"> • Primary: Get Started Free • Secondary: View Documentation 	TBD	
Update Sales collaterals	<ul style="list-style-type: none"> • Battlecards • Pitch Deck • Sell Sheet 	N/A	TBD	

Instrumentation

Operation	Event	Trigger	Properties

Event	Trigger	Properties
TM_TCTypeBDD	The user selects the test case type as 'BDD' from the drop down	user_id, project_id
"TM_TcCreatedNotification" (Existing)	When a BDD TC is created	Test_Template: BDD
"TM_AddResultCtaClicked" - Existing	When A BDD result is added	Test_Template: BDD
"TM_CiConfirmImportCtaClicked" - Existing	Currently -This is event is fired when a user clicks to confirm CSV import. Similarly this should be fired everytime the users clicks to start the Feature file import. (This is the last click - after which the import begins)	Import_type - CSV/Feature
TM_Export_Feature_Cta_clicked	To be fired when users clicks on export as feature file	default properties
TM_Export_Feature_Cta_clicked_in modal	To be fired when users clicks on export as feature file modal having the 2 radio options	Export_Option - Combined/Single

- **BE Instrumentation:**

- We should be able to see the No of Feature files exported/ imported at a given time
-

Success Metrics

- **No of BDD TCs created/exported/imported per month**
- No of results added to BDD TCS

References / Key MOMs

Decisions that need to be taken

1. Mapping Scenarios as Steps VS Mapping a Scenario as a Test Case

2. **Examples** - During execution, do we want to allow multiple instances of a scenario based on the examples (Similar to what we have for configuration)

3. Feature Files

4. Automation - TM_TO

Competitor	Scenarios as Steps VS Mapping a Scenario as a Test Case	Examples

Test Rail	Maps Scenarios as steps	Allows in RTE, does not consider diff instances during execution
Qtest	Provided a BDD editor inside steps	Allows in RTE, does not consider diff instances during execution
TestMo	Provided a BDD editor inside steps	Allows in RTE, does not consider diff instances during execution
Testomat	If a TC has more than 1 scenario, it will only take the 1st Scenario	Allows in RTE, consider diff instances during execution
Xray	One Scenario field per TC	Allows, consider diff instances during execution
Zephyr	One Scenario field per TC	Allows, does not consider diff instances during execution

Data points from QA folks and customers (Yash, Aprit and Yusuf (Holidu QA))

1. **Scenario-Level Mapping:** Each scenario corresponds to a test case level, while the feature block corresponds to a folder-level structure.
2. **Naming Conventions:** Scenario names should align with test case names, similar to how “IT” blocks have names.
3. **Export Challenges:** Exporting multiple scenario feature files or single files is a pain point, as it results in excessive file clubbing.
 - This was also highlighted by **Holidu (Zephyr Scale customer)** as a key issue they want to solve—combining multiple scenario exports into a single feature file.
4. **Flexibility in Feature Part & Background:** Allowing users to name the feature part and background will enhance usability.
5. **Phase Two Consideration for Examples:**
 - Yash noted that BDD examples weren’t being used; explicit values were present instead.
 - Since the Holidu **deal blocker doesn’t explicitly require** BDD examples, they are currently planned for **Phase Two**.
6. **Manual & Automation Use Cases:**
 - Test cases are often authored in the test management tool first and manually executed before automation is implemented.
 - The need for manual vs. automated execution depends on the company’s automation maturity level.

Separate Fields VS one Gherkin Field ☰

Factor	Approach 1: Separate Fields (Feature, Background, Scenario as distinct fields)	Approach 2: Combined Gherkin Editor (Single structured field for Feature, Background, Scenario)
--------	--	--

Alignment with Current Test Case Structure (One Scenario per Test Case)	High – Clearly separates elements like Feature, Background, and Scenario as structured fields	Low – Even though it mimics how a feature file would look like, but this might give the idea that a test case can have multiple scenarios like a feature file
UI Simplicity & Design Elegance	Low-Medium – More UI fields increase complexity, but structured clarity	High – Single field reduces UI clutter,
Content Flexibility (Images, Formatting, RTE Support)	High – RTE for Background allows inserting images, formatted text, etc.	Low-Medium – Gherkin editor is text-based; no native RTE support for images
Data Structuring & Field-Level Validation	High – Each field can have independent validation and structured storage. Feature is a mandatory field which needs to be validated	Medium – Needs full-text parsing to extract and validate sections, cannot put validations on feature, which is a must have field in the TC
Import Handling (Feature Files → Test Cases)	High – Direct mapping of Feature → Folder, Scenario → Test Case	Medium – Parsing required to extract separate scenarios
Export Handling (Test Cases → Feature Files)	Medium – Requires merging logic but ensures structured export	Low-Medium – For multiple TCs export, parsing the Features and backgrounds, might have its own set of issues
Scalability for Future Enhancements (Examples, Test Runs, etc.)	Medium – Needs additional mapping when introducing Examples	High – Examples fit naturally into the structured format
Engineering Complexity (Parsing, Storage, Processing)	High – Easier storage, retrieval, and validation as fields are separate	Medium – Parsing complexity increases, as sections need to be extracted dynamically
Collaboration & Version Control	High – Easier tracking of changes at the field level	Medium – Changes to one section (e.g., Scenario) require modifying the entire feature